FUZZY DATA MINING AND GENETIC ALGORITHMS APPLIED TO INTRUSION DETECTION

Susan M. Bridges, Associate Professor
Rayford B. Vaughn, Associate Professor
Department of Computer Science
Mississippi State University
Box 9637
Mississippi State, MS 39762
(662) 325-2756 fax: (662) 325-8997
<vaughn,bridges>@cs.msstate.edu

Abstract

We are developing a prototype intelligent intrusion detection system (IIDS) to demonstrate the effectiveness of data mining techniques that utilize fuzzy logic and genetic algorithms. This system combines both anomaly based intrusion detection using fuzzy data mining techniques and misuse detection using traditional rule-based expert system techniques. The anomaly-based components are developed using fuzzy data mining techniques. They look for deviations from stored patterns of normal behavior. Genetic algorithms are used to tune the fuzzy membership functions and to select an appropriate set of features. The misuse detection components look for previously described patterns of behavior that are likely to indicate an intrusion. Both network traffic and system audit data are used as inputs for both components.

1. Introduction

The wide spread use of computer networks in today's society, especially the sudden surge in importance of e-commerce to the world economy, has made computer network security an international priority. Since it is not technically feasible to build a system with no vulnerabilities, intrusion detection has become an important area of research. Intrusion detection approaches are commonly divided into two categories: misuse detection and anomaly detection [1]. The misuse detection approach attempts to recognize attacks that follow intrusion patterns that have been recognized and reported by experts. Misuse detection systems are vulnerable to intruders who use new patterns of behavior or who mask their illegal behavior to deceive the detection system. Anomaly detection methods were developed to counter this problem. With the anomaly detection approach, one represents patterns of normal behavior, with the assumption that an intrusion can be identified based on some deviation from this normal behavior. When such a deviation is observed, an intrusion alarm is produced.

Artificial intelligence (AI) techniques have been applied to both misuse detection and anomaly detection. Rule based expert systems have served as the basis for several systems including SRI's Intrusion Detection Expert System (IDES)[2]. These systems encode an expert's knowledge of known patterns of attack and system vulnerabilities as if-then rules. The acquisition of these rules is a tedious and error-prone process; this problem (known as the knowledge acquisition bottleneck in expert system literature) has generated a great deal of interest in the application of machine learning techniques to automate the process of learning the patterns. Examples include the Time-based Inductive Machine (TIM) for intrusion detection [3] that learns sequential patterns and neural network-based intrusion detection systems [4]. More recently, techniques from the data mining area (mining of association rules and frequency episodes) have been used to mine normal patterns from audit data [5, 10, 15].

Problems are encountered, however, if one derives rules that are directly dependent on audit data [6]. An intrusion that deviates only slightly from a pattern derived from the audit data may not be detected or a small change in normal behavior may cause a false alarm. We have addressed this problem by integrating fuzzy logic with data mining methods for intrusion detection.

Fuzzy logic is appropriate for the intrusion detection problem for two major reasons. First, many quantitative features are involved in intrusion detection. SRI's Next-generation Intrusion Detection Expert System (NIDES) categorizes security-related statistical measurements into four types: ordinal, categorical, binary categorical, and linear categorical [2]. Both ordinal and linear categorical measurements are quantitative features that can potentially be viewed as fuzzy variables. Two examples of ordinal measurements are the CPU usage time and the connection duration. An example of a linear categorical measurement is the number of different TCP/UDP services initiated by the same source host. The second motivation for using fuzzy logic to address the intrusion detection problem is that security itself includes fuzziness. Given a quantitative measurement, an interval can be used to denote a normal value. Then, any values falling outside the interval will be considered anomalous to the same degree regardless of their distance to the interval. The same applies to values inside the interval, i.e., all will be viewed as normal to the same degree. The use of fuzziness in representing these quantitative features helps to smooth the abrupt separation of normality and abnormality and provides a measure of the degree of normality or abnormality of a particular measure.

We describe a prototype intelligent intrusion detection system (IIDS) that is being developed to demonstrate the effectiveness of data mining techniques that utilize fuzzy logic. This system combines two distinct intrusion detection approaches: 1) anomaly based intrusion detection using fuzzy data mining techniques, and 2) misuse detection using traditional rule-based expert system techniques. The anomaly-based components look for deviations from stored patterns of normal behavior. The misuse detection components look for previously described patterns of behavior that are likely to indicate an intrusion. Both network traffic and system audit data are used as inputs. We are also using genetic algorithms to 1) tune the fuzzy membership functions to improve

performance, and 2) select the set of features available from the audit data that provide the most information to the data mining component.

2. System Goals and Preliminary Architecture

Our long term goal is to design and build an intelligent intrusion detection system that is accurate (low false negative and false positive rates), flexible, not easily fooled by small variations in intrusion patterns, adaptive in new environments, modular with both misuse and anomaly detection components, distributed, and real-time. The architecture shown in Figure 1 has been developed with these goals in mind.

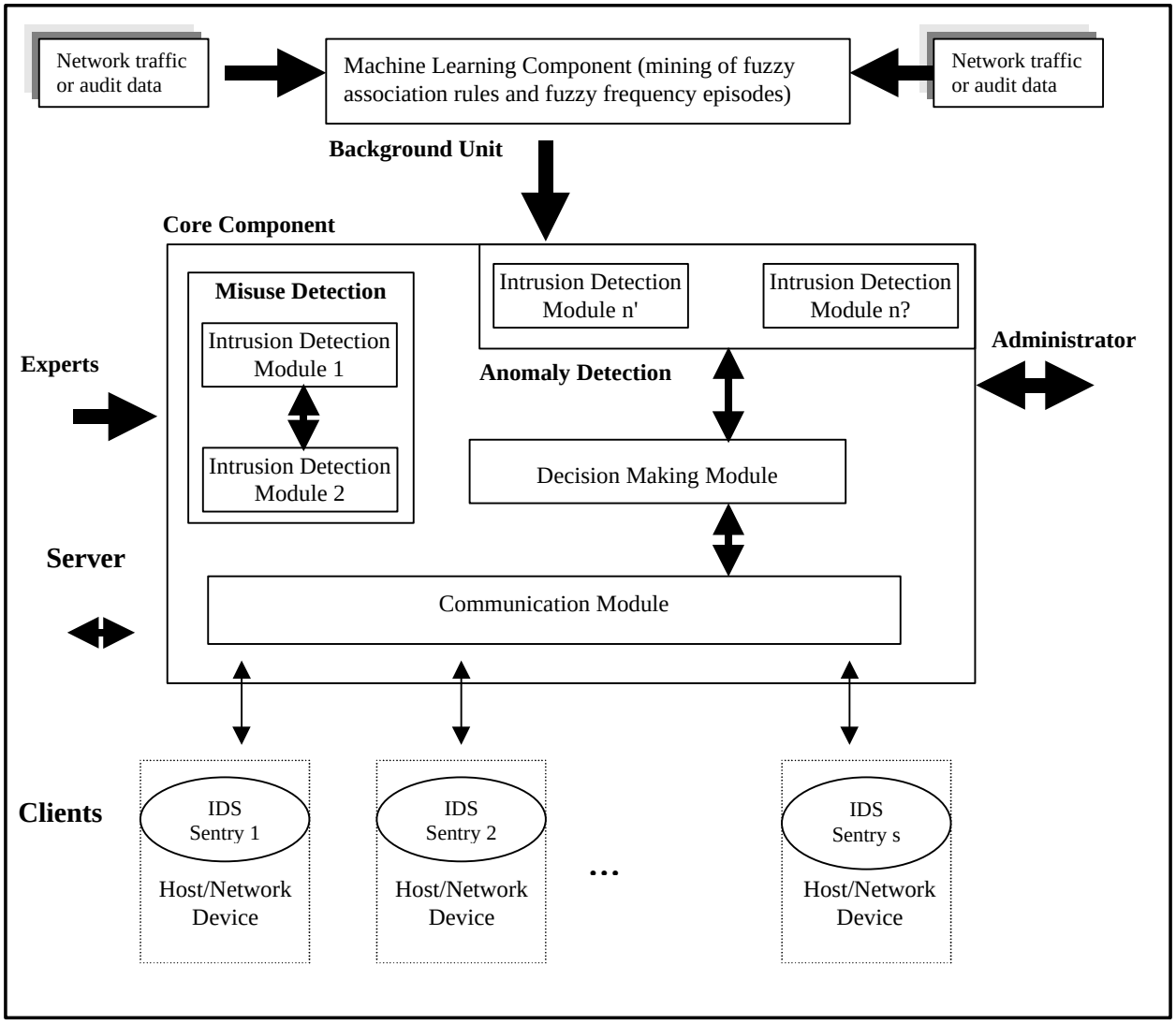


Figure 1: Architecture of IIDS

The Machine Learning Component integrates fuzzy logic with association rules and frequency episodes to "learn" normal patterns of system behavior. This normal behavior is stored as sets of fuzzy association rules and fuzzy frequency episodes. The *Anomaly*

Intrusion Detection Module extracts patterns for an observed audit trail and compares these new patterns with the "normal" patterns. If the similarity of the sets of patterns is below a specified threshold, the system alarms an intrusion. *Misuse Intrusion Detection Modules* use rules written in FuzzyCLIPS to match patterns of known attacks or patterns that are commonly associated with suspicious behavior to identify attacks. The use of fuzzy logic in both of these modules makes the rules of the system more flexible and less brittle. The machine learning component allows the system to adapt to new environments. The detection methods will be implemented as a set of intrusion detection modules. An intrusion detection module may address only one or even a dozen types of intrusions. Several intrusion detection modules may also cooperate to detect an intrusion in a loosely coupled way since these detection modules are relatively independent. Different modules may use different methods. For instance, one module can be implemented as a rule-based expert system and another module can be constructed as a neural network classifier. On the whole, this modular structure will ease future system expansion The *Decision-Making Module* will both decide whether or not to activate an intrusion detection module (misuse or anomaly) and integrate evaluation results provided by the intrusion detection modules. The Communication Module is the bridge between the intrusion detection sentries and the decision-making module. *Intrusion detection sentries* pre-process audit data and send results to the communication module. Feedback is returned to the sentries.

The architecture in Figure 1 is now under initial construction. Our preliminary results demonstrate that the fuzzy data mining techniques provide an effective means to learn and alert based on patterns extracted from large amounts of data. Our results also demonstrate that the integration of fuzzy logic with the data mining techniques enables improved performance over similar techniques that do not use fuzzy logic.

3. Anomaly Detection via Fuzzy Data Mining

We are combining techniques from fuzzy logic and data mining for our anomaly detection system. The advantage of using fuzzy logic is that it allows one to represent concepts that could be considered to be in more than one category (or from another point of view—it allows representation of overlapping categories). In standard set theory, each element is either completely a member of a category or not a member at all. In contrast, fuzzy set theory allows partial membership in sets or categories. The second technique, data mining, is used to automatically learn patterns from large quantities of data. The integration of fuzzy logic with data mining methods helps to create more abstract and flexible patterns for intrusion detection.

3.1 Fuzzy Logic

In the intrusion detection domain, we may want to reason about a quantity such as the number of different destination IP addresses in the last 2 seconds. Suppose one wants to write a rule such as

> **If** the number different destination addresses during the last 2 seconds was high
> **Then** an unusual situation exists.

Using traditional logic, one would need to decide which values for the number of destination addresses fall into the category high. As shown in Figure 2a, one would typically divide the range of possible values into discrete buckets, each representing a different set. The y-axis shows the degree of membership of each value in each set. The value 10, for example is a member of the set *low* to the degree 1 and a member of the other two sets, *medium* and *high*, to the degree 0. In fuzzy logic, a particular value can have a degree of membership between 0 and 1 and can be a member of more than one fuzzy set. In Figure 2b, for example, the value 10 is a member of the set *low* to the degree 0.4 and a member of the set *medium* to the degree 0.75. In this example, the membership functions for the fuzzy sets are piecewise linear functions. Using fuzzy logic terminology, the number of destination ports is a fuzzy variable (also called a linguistic variable), while the possible values of the fuzzy variable are the fuzzy sets *low*, *medium*, and *high*. In general, fuzzy variables correspond to nouns and fuzzy sets correspond to adjectives.

In our work, we are using the fuzzy logic system, FuzzyCLIPS [7] to represent patterns using a rule-based system. FuzzyCLIPS, developed by the National Research Council of Canada, is a fuzzy extension of the popular CLIPS expert system shell developed by NASA. FuzzyCLIPS provides several methods for defining fuzzy sets; we are using the three standard S, PI, and Z functions described by Zadeh [16]. The graphical shapes and formal definitions of these functions are shown in Figure 3. Each function is defined by exactly two parameters

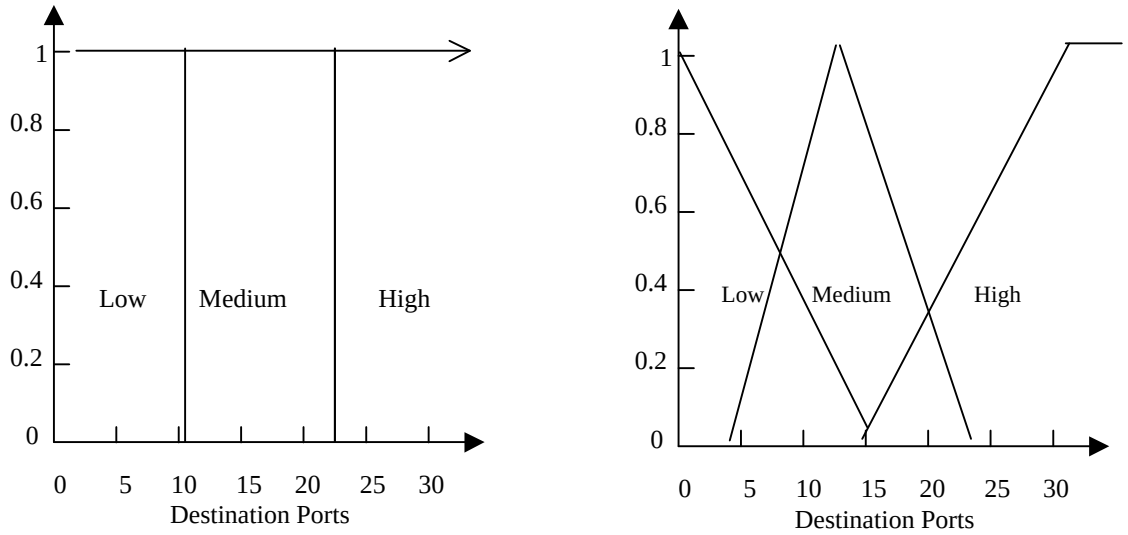
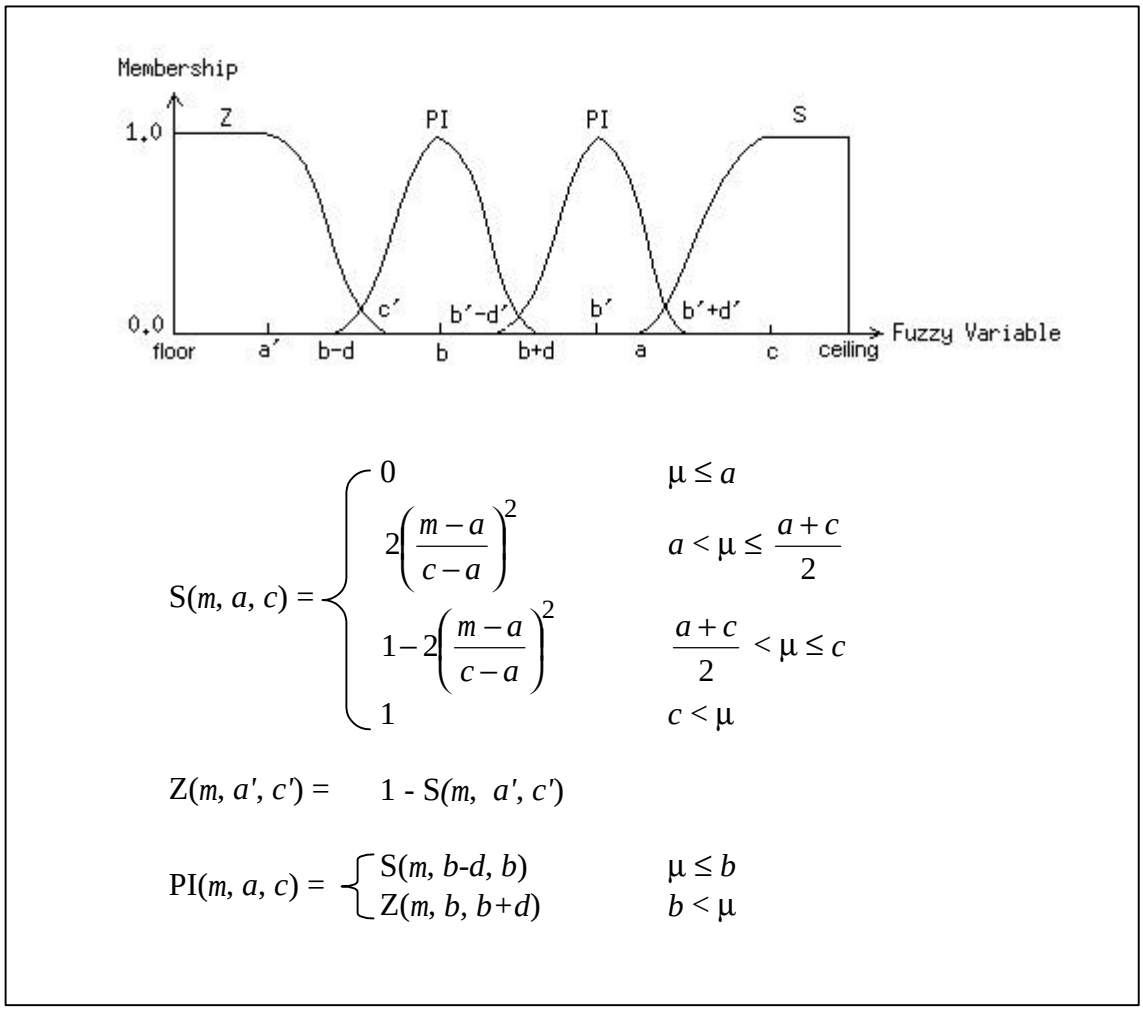


Figure 2: Non-fuzzy and fuzzy representations of sets for quantitative variables. The x-axis is the value of a quantitative variable. The y-axis is the degree of membership in the sets *low*, *medium*, and *high*.



$$S(m, a, c) = \begin{cases} 0 & \mu \le a \\ 2\left(\frac{m-a}{c-a}\right)^2 & a < \mu \le \frac{a+c}{2} \\ 1-2\left(\frac{m-a}{c-a}\right)^2 & \frac{a+c}{2} < \mu \le c \\ 1 & c < \mu \end{cases}$$

$$Z(m, a', c') = 1 - S(m, a', c')$$

$$PI(m, a, c) = \begin{cases} S(m, b\text{-}d, b) & \mu \le b \\ Z(m, b, b+d) & b < \mu \end{cases}$$

Figure 3. Standard function representation of fuzzy sets

Using fuzzy logic, a rule like the one shown above could be written as

If the DP = high

Then an unusual situation exists

where DP is a fuzzy variable and high is a fuzzy set. The degree of membership of the number of destination ports in the fuzzy set high determines whether or not the rule is activated.

3.2 Data Mining Methods

Data mining methods are used to automatically discover new patterns from a large amount of data. Two data mining methods, association rules and frequency episodes, have been used to mine audit data to find normal patterns for anomaly intrusion detection [5].

3.2.1 Association Rules

Association rules were first developed to find correlations in transactions using retail data [8]. For example, if a customer who buys a soft drink (A) usually also buys potato chips (B), then potato chips are associated with soft drinks using the rule A → *B.* Suppose that 25% of all customers buy both soft drinks and potato chips and that 50% of the customers who buy soft drinks also buy potato chips. Then the degree of support for the rule is s = 0.25 and the degree of confidence in the rule is c = 0.50. Agrawal and Srikant [8] developed the fast Apriori algorithm for mining association rules. The Apriori algorithm requires two thresholds of *minconfidence* (representing minimum confidence) and *minsupport* (representing minimum support). These two thresholds determine the degree of association that must hold before the rule will be mined.

3.2.2 Fuzzy Association Rules

In order to use the Apriori algorithm of Agrawal and Srikant [8] for mining association rules, one must partition quantitative variables into discrete categories. This gives rise to the "sharp boundary problem" in which a very small change in value causes an abrupt change in category. Kuok, Fu, and Wong [9] developed the concept of fuzzy association rules to address this problem. Their method allows a value to contribute to the support of more than one fuzzy set (see [10] for details). We have modified the algorithm of Kuok, Fu, and Wong [9], by introducing a normalization factor to ensure that every transaction is counted only one time. An example of a fuzzy association rule mined by our system from one set of audit data is:

$$\{ \text{SN}=\textit{LOW}, \text{FN}=\textit{LOW} \} \rightarrow \{ \text{RN}=\textit{LOW} \}, \qquad c = 0.924, s = 0.49$$

where SN is the number of SYN flags, FN is the number of FIN flags and RN is the number of RST flags in a 2 second period.

When presented with a set of audit data, our system will mine a set of fuzzy association rules from the data. These rules will be considered a high level description of patterns of behavior found in the data. For anomaly detection, we mine a set of rules from a data set with no intrusions (termed a reference data set) and use this as a description of normal behavior. When considering a new set of audit data, a set of association rules is mined from the new data and the similarity of this new rule set and the reference set is computed. If the similarity is low, then the new data will cause an alarm. Figure 4 shows results from one experiment comparing the similarities with the reference set of rules mined from data without intrusions and with intrusions. It is apparent that the set of rules mined from data with no intrusions (baseline) is more similar to the reference rule set than the sets of rules mined from data containing intrusions.

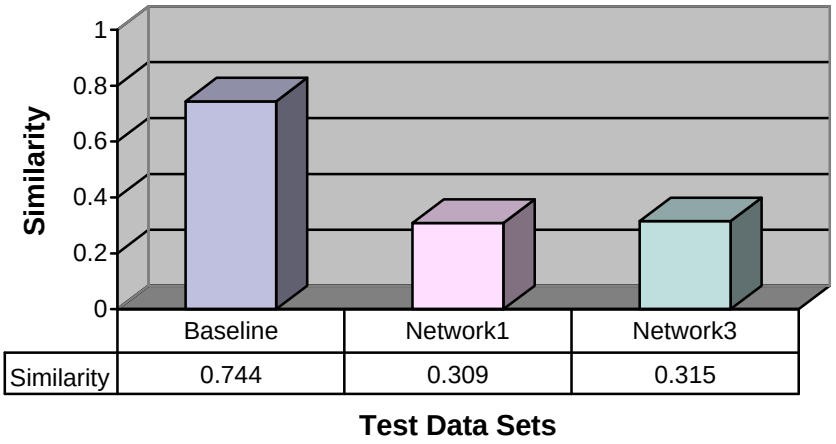


Figure 4. Comparison of Similarities Between Training Data Set and Different Test Data Sets for Fuzzy Association Rules (minconfidence=0.6; minsupport=0.1
Training Data Set: reference (representing normal behavior)
Test Data Sets: baseline (representing normal behavior),
network1 (including simulated IP spoofing intrusions), and
network3 (including simulated port scanning intrusions)

3.2.3 Frequency Episodes

Mannila and Toivonen [11] proposed an algorithm for discovering simple serial frequency episodes from event sequences based on minimal occurrences. Lee, Stolfo, and Mok [5] have applied this method to the problem of characterizing frequent temporal patterns in audit data. We have modified the method of Mannila and Toivonen to mine to fuzzy frequency episodes. In Mannila and Toivonen's method [11], an event is characterized by a set of attributes at a point in time. An episode $P(e_1,e_2, \ldots, e_k)$ is a sequence of events that occurs within a time window $[t,t']$. The episode is minimal if there is no occurrence of the sequence in a subinterval of the time interval. Given a threshold of *window* (representing timestamp bounds), the frequency of $P(e_1,e_2, \ldots, e_k)$ in an event sequence $\boldsymbol{S}$ is the total number of its minimal occurrences in any interval smaller than *window*. So, given another threshold *minfrequency* (representing minimum frequency), an episode $P(e_1,e_2, \ldots, e_k)$ is called frequent, if $frequency(P)/n \geq minfrequency$.

3.2.4 Fuzzy Frequency Episodes

We have developed a method for integrating fuzzy logic with frequency episodes [10]. The need to develop fuzzy frequency episodes comes from the involvement of quantitative attributes in an event. Other than the difference in calculating the frequency (or minimal occurrence) of an episode, our algorithm is similar to Mannila and Toivonen's algorithm [11] for mining frequency episodes. An example of a fuzzy frequency episode rule mined by our system is given below:

{ E1: PN=*LOW*, E2: PN=*MEDIUM* } →{ E3: PN=*MEDIUM* },
c = 0.854, s = 0.108, w = 10 seconds

where E1, E2, and E3 are events that occur in that order and PN is the number of distinct destination ports within a 2 second period.

Similarity results very much like those obtained for fuzzy association rules (Figure 4 for example) were obtained for fuzzy frequency episodes. In addition, we also compared the false positive rate for identifying intrusions obtained when non-fuzzy frequency episode rules were used with those obtained when fuzzy frequency episodes were used. These results demonstrate that the use of fuzzy logic with frequency episodes results in a reduction of the false positive error rate.

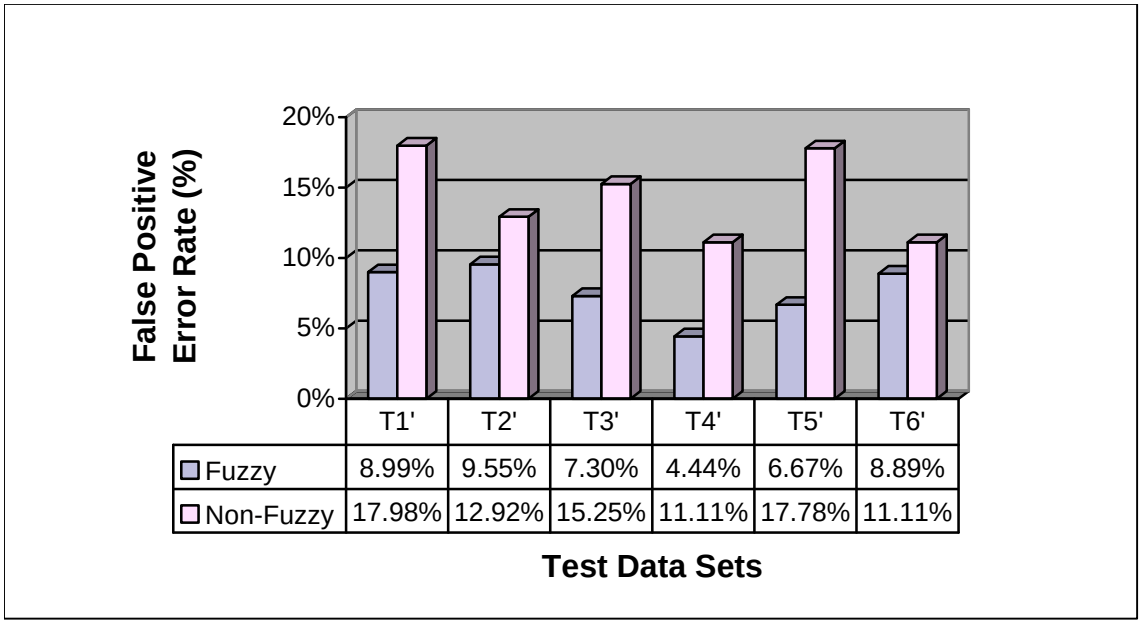


Figure 5: Comparison of False Positive Error Rates of Fuzzy Episode Rules and Non-Fuzzy Episode Rules
Fuzzy episode rule parameters: minconfidence=0.8; minsupport=0.1; minoccurrence=0.3; window=15s
Non-fuzzy episode rule parameters: minconfidence=0.8; minsupport=0.1; window=15s
Training Data Set: 3 hour training data (representing normal behavior)
Test Data Set: T1', T2', T3' (representing normal behavior), and T4', T5', T6' (include simulated *mscan* intrusions)

4. Misuse Detection Components

The misuse detection components are small rule-based expert systems that look for known patterns of intrusive behavior. The FuzzyCLIPS system allows us to implement both fuzzy and non-fuzzy rules. A simple example of a rule from the misuse detection component is given below:

IF the number of consecutive logins by a user is greater than 3
THEN the behavior is suspicious

Information from a number of misuse detection components will be combined by the decision component to determine if an alarm should be result.

5. Genetic Algorithms

Genetic algorithms are search procedures often used for optimization problems. When using fuzzy logic, it is often difficult for an expert to provide "good" definitions for the membership functions for the fuzzy variables. We have found that genetic algorithms can be successfully used to tune the membership functions of the fuzzy sets used by our intrusion detection system [13]. Each fuzzy membership function can be defined using two parameters as shown in Figure 3. Each chromosome for the GA consists of a sequence of these parameters (two per membership function). An initial population of chromosomes is generated randomly where each chromosome represents a possible solution to the problem (an set of parameters). The goal is to increase the similarity of rules mined from data without intrusions and the reference rule set while decreasing the similarity of rules mined from intrusion data and the reference rule set. A fitness function is defined for the GA which rewards a high similarity of normal data and reference data while penalizing a high similarity of intrusion data and reference data. The genetic algorithm works by slowly "evolving" a population of chromosomes that represent better and better solutions to the problem.

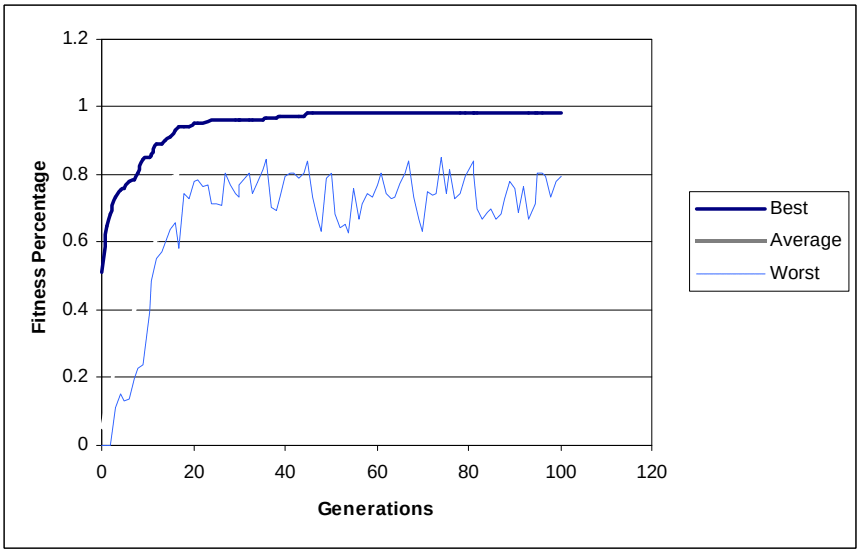


Figure 6 The evolution process of the fitness of the population, including the fitness of the most fit individual, the fitness of the least fit individual and the average fitness of the whole population

Figure 6 shows how the value of the fitness function changes as the GA progresses. The top line represents the fitness (or quality of solution) of the best individual in the population. We always retain the best individual from one generation to the next, so the fitness value of the best individual in the population never decreases. The middle line, showing the average fitness of the population, demonstrates that the overall fitness of the population continues to increase until it reaches a plateau. The lower line, the fitness of the least fit individual, demonstrates that we continue to introduce variation into the population using the genetic operators of mutation and crossover. Figure 7 demonstrates the evolution of the population of solutions in terms of the two components of the fitness function (similarity of mined ruled to the "normal" rules and similarity of the mined rules to the "abnormal" rules.) This graph also demonstrates that the quality of the solution increases as the evolution process proceeds.

It is often difficult to know which items from an audit trail will provide the most useful information for detecting intrusions. The process of determining which items are most useful is called *feature selection* in the machine learning literature. We have conducted a set of experiments in which we are using genetic algorithms both to select the measurements from the audit trail that are the best indicators for different classes of intrusions and to "tune" the membership functions for the fuzzy variables [14]. Figure 8 compares results when rules are mined 1) when there was no optimization and no feature selection, 2) when there was only optimization, and 3) when there was both optimization and feature selection. These results demonstrate that the GA can effectively select a set of features for intrusion detection while it tunes the membership functions. We have also found that the GA can identify different sets of features for different types of intrusions [14].

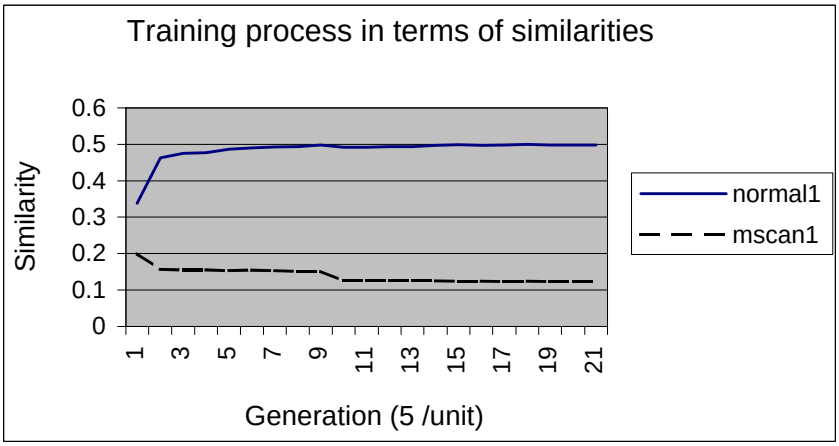


Figure 7: The evolution process for tuning fuzzy membership functions in terms of similarity of data sets containing intrusions (mscan1) and not containing intrusions (normal1) with the reference rule set.

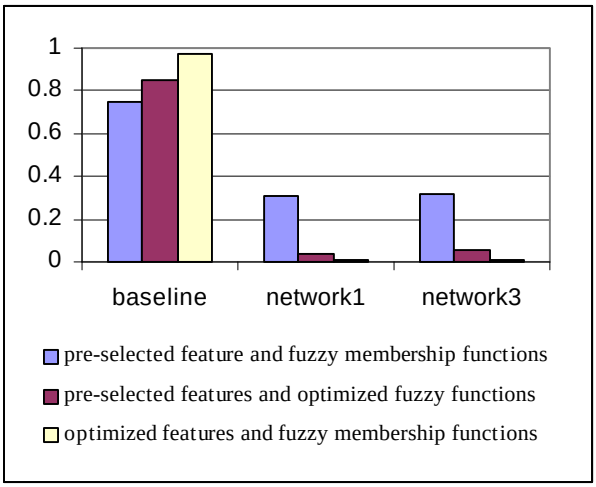


Figure 8: Comparison of the similarity results using 1) features and fuzzy membership functions selected by the expert, 2) features selected by the expert and membership functions optimized by a GA, and 3) features selected by the GA and membership functions optimized by the GA.

6. Summary and Future Work

We have integrated data mining techniques with fuzzy logic to provide new techniques for intrusion detection. Our system architecture allows us to support both anomaly detection and misuse detection components at both the individual workstation level and at the network level. Both fuzzy and non-fuzzy rules are supported within the system. We have also used genetic algorithms to tune the membership functions for the fuzzy variables used by our system to and select the most effective set of features for particular types of intrusions.

We are currently building misuse detection components, the decision module, additional machine learning components, and a graphical user interface for the system. Also under investigation, are possible solutions to the problem of dealing with "drift" in normal behavior. We plan to extend this system to operate in a high performance cluster computing environment.

References

1. Allen, J., Alan Christie, Willima Fithen, John McHugh, Jed Pickel, Ed Stoner. 2000. *State of the Practice of Intrusion Detection Technologies*. CMU/SEI-99-TR-028. Carnegie Mellon Software Engineering Institute. (Downloaded from http://sei.cmu.edu/publications/documents/99.reports/99tr028abstract.html).

2. Lunt, T. 1993. Detecting intruders in computer systems. In *Proceedings of 1993 conference on auditing and computer technology*. (Downloaded from http://www2.csl.sri.com/nides/index5.html on 3 February 1999.)

3. Teng, H., K. Chen, and S. Lu. 1990. Adaptive real-time anomaly detection using inductively generated sequential patterns. In *Proceedings of 1990 IEEE computer society symposium on research in security and privacy held in Oakland, California, May 7-9, 1990*, by IEEE Computer Society, 278-84. Los Alamitos, CA: IEEE Computer Society Press.

4. Debar, H., M. Becker, and D. Siboni. 1992. A neural network component for an intrusion detection system. In *Proceedings of 1992 IEEE computer society symposium on research in security and privacy held in Oakland, California, May 4-6, 1992*, by IEEE Computer Society, 240-50. Los Alamitos, CA: IEEE Computer Society Press.

5. Lee, W., S. Stolfo, and K. Mok. 1998. Mining audit data to build intrusion detection models. In *Proceedings of the fourth international conference on knowledge discovery and data mining held in New York, New York, August 27-31, 1998*, edited by Rakesh Agrawal, and Paul Stolorz, 66-72. New York, NY: AAAI Press.

6. Ilgun, K., and A. Kemmerer.1995. State transition analysis: A rule-based intrusion detection approach. *IEEE Transaction on Software Engineering* 21(3): 181-99.

7. Orchard, R. 1995. *FuzzyCLIPS version 6.04 user's guide.* Knowledge System Laboratory, National Research Council Canada.

8. Agrawal, R., and R. Srikant. 1994. Fast algorithms for mining association rules. In *Proceedings of the 20th international conference on very large databases held in Santiago, Chile, September 12-15, 1994*, 487-99. San Francisco, CA: Morgan Kaufmann. (Downloaded from http://www.almaden.ibm.com/cs/people/ragrawal/papers/vldb94_rj.ps on February 1999.)

9. Kuok, C., A. Fu, and M. Wong. 1998. Mining fuzzy association rules in databases. *SIGMOD Record* 17(1): 41-6. (Downloaded from http://www.acm.org/sigs/sigmod/record/ issues/9803 on 1 March 1999).

10. Luo, J. 1999. *Integrating fuzzy logic with data mining methods for intrusion detection*. M.S. Thesis, Mississippi State University.

11. Mannila, H., and H. Toivonen. 1996. Discovering generalized episodes using minimal occurrences. In *Proceedings of the second international conference on knowledge discovery and data mining held in Portland, Oregon, August, 1996*, by AAAI Press, 146-51. (Downloaded from http://www.cs.Helsinki.FI/research/fdk/ datamining/pubs on 19 February 1999.)

12. Porras, P., and A. Valdes. 1998. Live traffic analysis of TCP/IP gateways. In *Proceedings of the 1998 ISOC symposium on network and distributed systems security held in March, 1998.* (downloaded from http://www2.csl.sri.com/emerald/downloads.html on 1 March 1999.)

13. Wang, W., and S. Bridges. 1999. Genetic algorithm optimization of membership functions for mining fuzzy association rules. Submitted for publication to the 7th International Conference on Fuzzy Theory and Technology (FT&T 2000)

14. Shi, Fajun, Susan M. Bridges, Rayford B. Vaughn 2000. The Application of Genetic Algorithms for Feature Selection in Intrusion Detection. In preparation.

15. Mukkamala, R., J. Gagnon, and S. Jajodia. 2000. Integrating data mining techniques with intrusion detection methods. In *Research Advances in Database and Information Systems Security*, Vijay Atluri and John Hale, editors, Kluwer Publishers, Boston, MA. 33-46.

16. Zadeh, L. A. 1973. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics,* SMC-3.